

# Release A CDR RID Report

**Date Last Modified** 10/9/95

**Originator** Hal Folts/Leon Jordan

**Phone No** 301  
286-3512/301-794-18

**Organization** ESDIS/CSC

**E Mail Address** folts@eos.nasa.gov/qjordan@ulabsgi.gsfc.nasa.gov

**Document** Release A MSS Design Specification (305-CD-013-001)

<b>RID ID</b>	CDR 80
<b>Review</b>	Release A CDR
<b>Originator Ref</b>	DSNO 5
<b>Priority</b>	2

**Section** Management Logistics

**Page** beginning Page 5-1

**Figure Table** N/A

~~CSC, Section 5~~

**Category Name** MSS Design

**Actionee** ECS

**Sub Category**

**Subject** Management Logistics Detail

## **Description of Problem or Suggestion:**

- (1) A detailed discussion and scenarios are needed that show how versions are managed. For example,
- How are software dependencies established and tracked? If custom S/W unit Y is needed between COTS packages X and Z, and then the version for Z changes, what reminds us to look at Y?
  - How are dependencies between S/W versions and test information (plans, procedures, scripts, test data, expected output, test output, and test analyses) established and tracked. When a version changes, it may be necessary to alter the test information.
  - How are versions of COTS units managed. COTS versions may be expected to change often.
- (2) A capability is needed for a configuration management discovery service to compare the versions that are actually running against the formally accepted baseline.
- What is to stop the operator from using a version of a unit that is different from the baseline (e.g., because of a problem, a previous version is used).
  - If this happens, how do we know?
- (3) More detail is needed in the textual description and scenarios in the following areas:
- Interactions among the logistics services (e.g., Change Manager and Change Request Manager for GetSoftwareAttribute request -- see Sect. 5.3.2, Page 5-21 last Parag. and Sect 5.3.3, Page 5-35, last Parag)
  - Detailed lists of parameters and other information in such things as "access profiles" (Sect 5.3.2, Page 5-21 middle Parag.), "attributes that describe software" (ibid. last Parag), contents of a GetVerification Request (ibid).
  - Specific details and scenarios that describe audit capability, levels of management (e.g., between the site and SMC), and the interface with CCBs.

## **Originator's Recommendation**

- (1) Supply the version management detailed descriptions and scenarios.
- (2) Supply the CM Discovery Service.
- (3) Supply the requested detailed descriptions and scenarios.

---

## **GSFC Response by:**

## **GSFC Response Date**

**HAIS Response by:** Gary Forman

**HAIS Schedule** 9/20/95

**HAIS R. E.** Benzell Floyd

**HAIS Response Date** 9/29/95

(1) Sections 5.3.1 will be changed to clarify version management capability. The following information is provided in response to the specific examples cited:

Software dependencies are established by the software developers and they are tracked through use of the Baseline Manager application. The Baseline Manager is used to maintain records that identify versions of hardware and software items that a baseline contains as well as record item interdependencies. In the example cited, if COTS package Z changes, one would query the Baseline Manager application to determine if other software is affected.

The Baseline Manager application will be used to record version identifiers of test plans, procedures, scripts, test data, etc. associated with deployed versions of software.

# Release A CDR RID Report

The Baseline Manager application will track and report the appropriate version of COTS software for each operational baseline just as it does for custom software. COTS versions are not expected to change frequently. But when a new version of COTS software is produced, ESDIS CCB approval (in accordance with ECS configuration management procedures) must be obtained before the new version is inserted into the operational baseline.

(2) There is no Release A requirement for a service to compare installed software versions against the formally accepted baseline. However, the Physical Configuration Manager application will provide a report (see Section 6.5.6.3) that list the software currently installed on a host. This list can be compared manually to a list that the Baseline Manager application can produce of the software that is supposed to be installed on that host (see reports in Section 5.5.1).

The following information is provided in response to the questions raised concerning unapproved installation of software: There is no automated capability that will "stop" an operator from using a version of a software unit that is not part of the current ESDIS approved operational baseline if he or she has access to the previous version and the means to install it. Operators are expected to follow published configuration management procedures that will provide instructions for operators to follow when confronted with a software problem. If an operator was to install a previous version of software (without appropriate approvals), detection will occur if the previous version causes problems or through comparison of reports obtained from the Physical Configuration Manager and the Baseline Manager applications.

(3) The DID 305 will be expanded to provide more specific details about the areas mentioned. The following information is provided in response to specific RID comments:

"GetSoftwareAttributes" is a ClearCase script that makes select information stored in ClearCase available in a format that can be used by Baseline Manager.

"Access Profiles" is an abstract object class, specified in figure 5.3-2 and para 5.3.2.1, that represents ClearCase's inherent, configurable capability to regulate who can perform various operations on software library files. Controls are implemented via mechanisms such as userid and group permissions, pre- and post-op triggers that query on library object and CCR status, and functions reserved for library object owners and designated administrators. ClearCase will be deployed configured with initial profile settings, most of which can be subsequently re-configured on-site by CM or system administrators. ClearCase manuals describe control capabilities and defaults.

DID 305 will be amplified to detail the specific "attributes that describe software".

GetVerification is a script that determines the existence of a CCR authorizing a change to a software library object. Its signature is identified in the Change Request Object Model (figure 5.3-3) and the Change Request Manager Class description (para 5.3.3.3). Clarification will be added to the DID 305.

Discussions and scenarios concerning site vs SMC levels of configuration management activity -- including audits -- are more appropriately detailed in DID 102, volume 2 (M&O CM Plan), DID 601 (M&O Management Plan), DID 604 (ECS Ops Concept), and DID 605 (M&O Procedures). Also, note that the configuration management application service does not "interface" with CCBs, per se. Rather, it is used to support CCB activities by maintaining configuration identification and status data and by aiding in the orderly processing and disposition of system change requests. For example, reports that detail baselined system configurations and identify their related documents can be used to support configuration audits for a CCB considering approving a change for production.

---

**Status Closed**

**Date Closed 10/9/95**

**Sponsor Folts**

\*\*\*\*\*

**Attachment if any**

\*\*\*\*\*

---